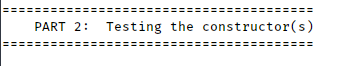
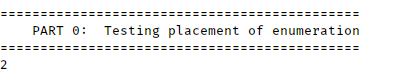
**CSCI 1107 Lab Experience Six**

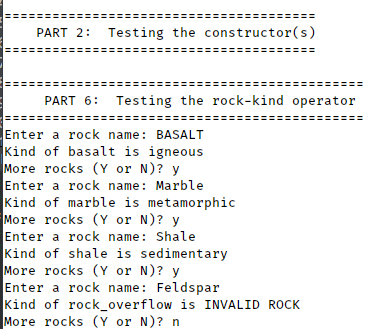
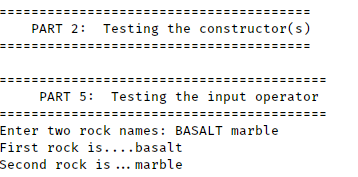
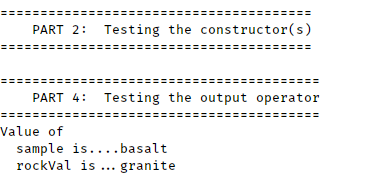
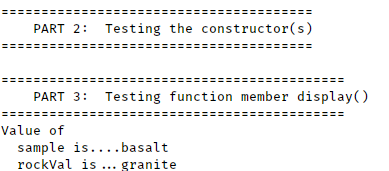
You are to do the following from your lab manual. If you use a word processor to reproduce the lab manual you need to also include the question with your answer.

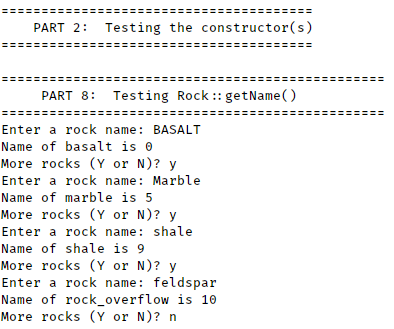
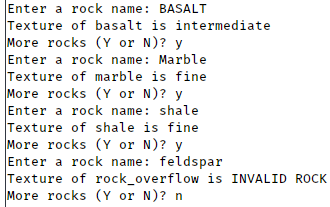
1. Do Lab 4.1 and Project 4.1 all parts on pages 51 – 64. On page 64 you should also overload the postfix operator. Recall the differences between the post and pre operations when used in calculations.

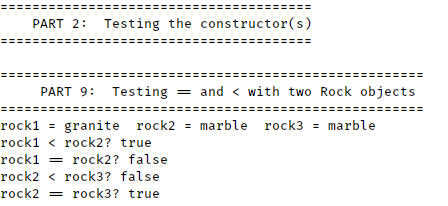
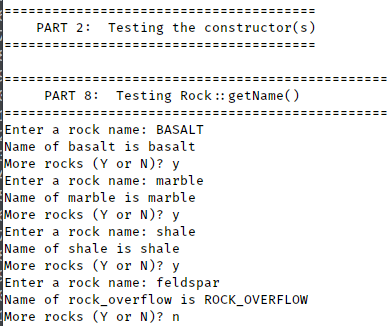
Capture the console window for each test run and paste it in your word document.

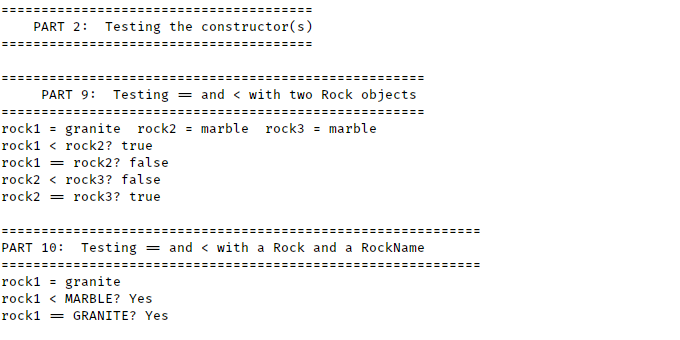
**(2 columns of screenshots, roughly in order of left→right→next line as they appear in the manual)**











/\* Project4-1.cpp

\* Generates a bar graph from a Rockfile.txt

\* Johnathan Lee CSCI 1107

\* Due 02/27/18

\*/

#include <fstream>

#include <iomanip>

#include <iostream>

#include "Rock.h"

using namespace std;

void **countOccurences**(int count[], string fileName);

void **barGraph**(int counts[]);

int **main**() {

int count[*ROCK\_OVERFLOW*] = {};

countOccurences(count, "Rockfile.txt");

barGraph(count);

return 0;

}

/// \brief Counts the occurences of rocks in a file.

/// \param count The array to store counts in, indexed by RockNames.

/// \param fileName the name of the file to read from.

/// \post count contains the number of times each Rock's human readable name

/// appears in fileName.

void **countOccurences**(int count[], string fileName) {

Rock rock;

ifstream inFile;

inFile.open(fileName);

// Count occurences

while (inFile >> rock) {

count[rock.getName()]++;

/\* DEBUG STUFF

cout << "Read rock: " << rock

<< "\nCount is now: " << count[rock.getName()];

\*/

}

}

/// \brief Outputs a bargraph of the counts of rocks.

/// \param counts The array containing the counts, indexed by RockNames.

/// \post A bar graph has been printed in the following format for each rock:

/// <ROCK NAME>: <ONE 'X' FOR EACH COUNT> (<INTEGER COUNT>)

/// ... <MORE OF THE ABOVE FOR EACH ROCK>

void **barGraph**(int counts[]) {

// Start at BASALT(0).

for (Rock rock = *BASALT*; rock < *ROCK\_OVERFLOW*; ++rock) {

cout << setw(20) << rock << ":";

for (int i = 0; i < counts[rock.getName()]; i++) {

cout << "X";

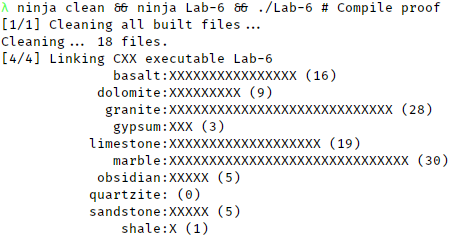
}

cout << " (" << counts[rock.getName()] << ")" << endl;

}

}

Screenshot of the output for the provided Rockfile.txt with compilation proof:



1. ~~Do problem 14 on page 191.~~

~~Capture the console window for each test run and paste it in your word document.~~

**Skipped until next lab, as per your email.**

**What to turn in:**

1. Printouts of your word document
2. Printouts of your source code.
3. A hard copy of the questions and answers from your lab manual.
4. Compress the word document and the source code from all of your programs and submit them to the D2L assignment folder called Lab Experience Six.